DepthComp: Real-time Depth Image Completion Based on Prior Semantic Scene Segmentation

Amir Atapour-Abarghouei
amir.atapour-abarghouei@durham.ac.uk
Toby P. Breckon
toby.breckon@durham.ac.uk

Engineering and Computer Science
Durham University
Durham, UK

Abstract

We address plausible hole filling in depth images in a computationally lightweight methodology that leverages recent advances in semantic scene segmentation. Firstly, we perform such segmentation over a co-registered color image, commonly available from stereo depth sources, and non-parametrically fill missing depth values based on a multi-pass basis within each semantically labeled scene object. Within this formulation, we identify a bounded set of explicit completion cases in a grammar inspired context that can be performed effectively and efficiently to provide highly plausible localized depth continuity via a case-specific non-parametric completion approach. Results demonstrate that this approach has complexity and efficiency comparable to conventional interpolation techniques but with accuracy analogous to contemporary depth filling approaches. Furthermore, we show it to be capable of fine depth relief completion beyond that of both contemporary approaches in the field and computationally comparable interpolation strategies.

1 Introduction

Three dimensional scene understanding based on scene depth is becoming ever more applicable to areas such as autonomous driving, interactive entertainment, environment modeling and alike [15, 28, 32]. However, complete (hole-free) scene depth is not readily obtainable from conventional capture devices. Missing or invalid depth values are commonplace, resulting in the need for depth filling as a time-consuming special case facet of any subsequent processing.

Prior work has considered numerous approaches to complete color images successfully [1, 6, 16, 22, 24, 45]. However, due to the different nature of scene depth from color including the absence of granular texture, object separation and the in-scene transferability of varying depth sub-regions, color image inpainting is less effective within the depth modality.

In this paper, we propose a simple and efficient method for depth image completion that utilizes a prior semantic segmentation labeling of the accompanying color image [3]. The depth completion process is performed with reference to object boundaries on a pixel-wise basis in the depth image with reference to a language of holes, where pixel values are parsed to identify and fill instances of holes. The key contributions of this paper are:

- *Novelty* - an efficient and novel non-parametric strategy that preserves relief texture.
- *Efficiency and Accuracy* - more efficient and accurate than comparators (Section 4).
- *Reproducibility* - simple and effective algorithm that can be reproduced easily.

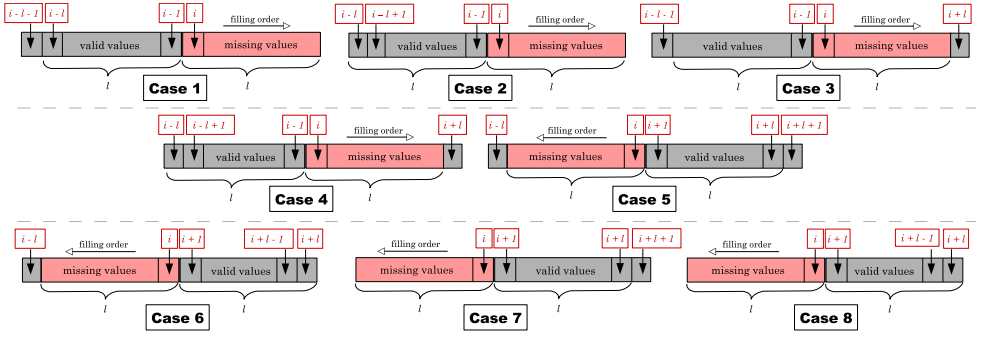

Figure 1: Exemplar constrained holes (row-wise), Cases 1-8.

2 Prior Work

Prior work in depth hole filling [2, 8, 9, 10, 35, 46] is not as extensive as color image completion and inpainting. Whilst many seminal color image completion techniques fall short when applied to depth maps [6, 16], there are specific depth filling techniques that leverage classic inpainting approaches, with or without modifications, to fill depth values [2, 23, 30, 51]. There have also been attempts to fill a target region in one of a set of multi-view photographs [4], to fill color and depth via depth-assisted texture synthesis [46], and a myriad of approaches utilizing filters [13, 14, 18, 34, 39, 41], temporal-based methods [6, 25, 38], reconstruction-based methods [17, 36, 47, 50], and others [2, 29, 35, 37, 40]. We focus on the most relevant to this work [29, 35, 40].

In a notable work, Liu *et al*. [35] improve upon the Fast Marching Method-based inpainting [45] for depth filling. By assuming that the adjacent pixels with similar color values are likely to have similar depth values as well, they introduce a *color term* into the weighting function to increase the contribution of the pixels with the same color.

By contrast, Qi *et al*. [40] use a fusion-based method integrated with a non-local filtering strategy. Their framework follows [17], utilizing a scheme similar to the non-local means scheme [11] to make more accurate depth predictions based on image textures.

Herrera *et al*. [29] propose depth inpainting guided by color assuming surfaces are continuous and smooth within their energy function. This smoothness term encourages flat depth planes in the completion process whilst ignoring the possibility of visible texture or relief in the filled region and hence limiting plausible completion characteristics.

Overall, prior work is characterized by a continuum from high-complexity with viable plausibility [2, 29, 35] to that of low complexity with limited plausibility, such as simple interpolation techniques. We propose a low complexity approach for plausible depth completion based on a proposed grammar of holes identified and filled on a row-wise basis.

3 Proposed Method

Our process uniquely leverages recent advances in semantic scene segmentation [3], such that completion can now be performed with reference to object boundaries within the scene. Here, focusing on the challenge of outdoor driving scenes, we utilize SegNet [3, 33], a deep convolutional architecture trained for urban scene segmentation in the context of vehicle autonomy. However, in general, any such approach that can perform accurate and efficient object or instance wise scene segmentation can suffice (illustrated in Figure 4).

Our technique is a computationally inexpensive completion approach requiring a maxi-

$l \leftarrow$ *length of the hole.*
$c \leftarrow$ *completion case identifier.*
if c *in* $\{1,2,3,4\}$ **then**
    $i \leftarrow$ *index of leftmost pixel in the hole*
else if c *in* $\{5,6,7,8\}$ **then**
    $i \leftarrow$ *index of rightmost pixel in the hole*
assign initial $v_0(i)$ according to case c
assign $slope$ according to case c
while i *is in the hole region* **do**
    update $v(i)$ according to case c
    if c *in* $\{1,2,3,4\}$ **then**
        $i \leftarrow i+1$.
    else if c *in* $\{5,6,7,8\}$ **then**
        $i \leftarrow i-1$.

Algorithm 1: Constrained Hole Completion

Type	%	Type	%
Case 1	11.19	Case 7	7.75
Case 2	0.32	Case 8	0.33
Case 3	57.02	Case 9	1.93
Case 4	1.99	Case 10	2.47
Case 5	3.44	Case 11	10.31
Case 6	0.22	Case 12	3.03
Filled	98.83	Unfilled	1.17

Table 1: Hole Frequency (KITTI [27])

	RMSE	PBMP	Run-time
Ours	0.4012	0.0021	97.38 *ms*

Table 2: Average RMSE, PBMP, & run-time (15 images form Middlebury [31]).

mum of three passes over the image on a row and column-wise basis. Within this context, a hole is now defined as a sequence of missing depth values constrained to one scene object within a single row/column of the depth image. To these ends, a depth hole (i.e. missing region in the image) is now comprised of multiple such *constrained holes*, all limited to the completion of a single row/column, with respect to a single adjacent scene object at the hole boundary. In a sense, we now have a grammar of holes, with each instance parsed based on row-wise adjacent depth availability for a consistent scene object.

As the image is scanned in raster order, each constrained hole discovered is identified as one of twelve possible completion cases with reference to both the pattern of missing depth and the consistency of surrounding segmented pixel labeling (Section 3.2). Whilst each case may be efficiently implemented in isolation, a common notion of informed re-sampling, behind the overall solution to all cases, provides underpinning plausible completion in the general sense. This avoids the simplicity of a brittle rule-based technique whilst taking advantage of a discrete set of hole occurrences at the local level to aid efficient implementation.

3.1 Semantic Segmentation

Here, we primarily use SegNet [3, 33] to perform the initial segmentation task. The SegNet architecture consists of encoder and decoder layers and a pixel-wise classification layer in a deep convolutional neural network (CNN) auto-encoder architecture. The encoder is similar to the convolutional layers in VGG16 [44], while the decoder maps the low-resolution feature maps from the encoder as inputs for subsequent pixel-wise semantic classification. Although SegNet [3] shows sufficient accuracy for our task, an *idealized* scene depth completion method requires absolute labeling accuracy beyond that of SegNet. As we illustrate (Figure 4), alternative segmentation models (object, instance or otherwise) [7, 19, 21, 43] can similarly be used, provided they produce limited mis-segmentation artefacts.

3.2 Hole Filling

Depth completion is performed in three images passes: primary row-wise, column-wise and secondary row-wise. For explanation purposes, we detail the outline of our approach solely in terms of image rows with the intermediate column-wise pass being purely a rotational analogue of the same. Each constrained depth hole can be identified as either one of eight non-parametrically solvable cases (subsequently outlined with a corresponding algorithmic

Method	RMSE	PBMP	Run-time
Linear Inter.	22.5868	0.2601	3.05 *ms*
Cubic Inter.	22.3810	0.2598	3.18 *ms*
GIF [35]	7.3281	0.2496	1.07*e*3 *ms*
SSI [29]	3.7970	0.1893	5.92*e*3 *ms*
FMM [45]	18.9501	0.2663	1.10*e*3 *ms*
EBI [1]	10.5448	0.1513	>1.2*e*5 *ms*
FBI [2]	0.8372	0.0863	>3.6*e*6 *ms*
Ours	0.8617	0.0917	3.83 *ms*

Table 3: RMSE, PBMP, & run-time (synthetic test image with ground truth depth).

Method	Example 1	Example 2
Linear Inter.	6.428 *ms*	7.643 *ms*
Cubic Inter.	6.998 *ms*	8.109 *ms*
GIF [35]	14.32*e*2 *ms*	16.14*e*2 *ms*
SSI [29]	20.4*e*3 *ms*	20.48*e*3 *ms*
FMM [45]	82.8*e*1 *ms*	83.5*e*1 *ms*
EBI [1]	>12*e*5 *ms*	>12*e*5 *ms*
FBI [2]	>36*e*5 *ms*	>36*e*5 *ms*
Ours	10.827 *ms*	11.516 *ms*
Ours+SegNet [3]	632.135 *ms*	633.091 *ms*

Table 4: Comparative run-times over KITTI dataset examples [27].

solution) or as one of four remaining unresolvable cases. When a case does not conform to a solvable case in a given pass, it is left to subsequent passes whereby the completion of other neighborhood pixels may allow subsequent resolution into one of these cases. In cases where a pixel remains unresolvable after all three passes, we refer to the use of linear or bilinear interpolation. From Table 1, we see the occurrence of these non-parametrically unresolvable cases is indeed very limited.

When a hole of a specific length is identified within a row, the information available to the left and the right of the hole within the same object boundaries is surveyed, and surface depth pattern is propagated into the hole region. A continuity coefficient (*slope*) is taken into account during this propagation to plausibly bridge the depth values on both sides of the hole. Although all constrained hole cases are essentially processed identically, the availability of valid depth values and appropriate sampling region govern the categorization of such row-wise constrained hole occurrences into a number of discrete cases. Of these twelve such completion cases, many are inherently similar in their characteristics with our detailed separation on a case-wise basis only aimed at maximizing accuracy and efficiency.

***Case 1*:** where the constrained hole ends at the rightmost boundary of the object, i.e. all depth values on the right side of the current object are missing, but the number of preceding depth values to the left of the hole exceeds the length of the hole itself.

$$v(i) = v(i-1) + [v(i-l) - v(i-l-1)] \times slope \tag{1}$$

Since such holes extend to the rightmost pixel in the current object, no depth information is available to the right of the hole, and as such there is no need to account for any in-filling continuity. Consequently, it suffices to identify the pattern of depth change to the left of the hole, the length of which is greater than the length of the hole itself, and propagate this pattern rightward, replicating the texture and relief detail present within the object boundary. As a result, $slope = 1$, and $v(i)$ is initialized to zero with updates as per Eqn. 1 with reference to Algorithm 1. See the illustration of Eqn. 1 terms in Figure 1 (Case 1).

***Case 2*:** where the constrained hole ends at the rightmost boundary the object (as per Case 1) but here, the number of preceding depth values to the left of the hole is exactly the same as the length of the hole itself.

$$v(i) = v(i-1) + [v(i-l+1) - v(i-l)] \times slope \tag{2}$$

Here we proceed as per Case 1, but with less depth information present to the left of the hole to identify and propagate any pattern rightward. As a result, $slope = 1$, and $v(i)$ is initialized to zero with updates as per Eqn. 2 with reference to Algorithm 1. See the illustration of Eqn. 2 terms in Figure 1 (Case 2).

***Case 3*:** where the constrained hole does not reach the leftmost or rightmost boundary edges of the scene object, i.e. the hole is contained within the object itself with valid depth

Method	Plastic (1270 × 1110)			Baby (1240 × 1110)			Bowling (1252 × 1110)		
	RMSE	PBMP	Run-time	RMSE	PBMP	Run-time	RMSE	PBMP	Run-time
Linear Inter.	1.3432	0.0229	28.036 *ms*	1.3473	0.0080	26.265 *ms*	1.4503	0.0430	21.081 *ms*
Cubic Inter.	1.2661	0.0212	30.488 *ms*	1.3384	0.0079	29.377 *ms*	1.4460	0.0418	23.685 *ms*
GIF [15]	0.7947	0.0331	31.08*e*2 *ms*	0.6008	0.0095	25.8*e*2 *ms*	0.9436	0.0412	48.75*e*2 *ms*
SSI [29]	1.7573	0.0102	42.36*e*3 *ms*	2.9638	0.0180	41.2*e*3 *ms*	6.4936	0.0455	71.12*e*3 *ms*
FMM [15]	0.9580	0.0435	93.93*e*1 *ms*	0.8349	0.0120	79.44*e*1 *ms*	1.2422	0.054	11.19*e*3 *ms*
EBI [9]	0.6952	0.0032	>36.*e*4 *ms*	0.6755	0.0024	>48*e*4 *ms*	0.4857	0.0035	>72*e*4 *ms*
FBF [9]	0.8643	0.0023	>10.8*e*6 *ms*	0.6238	0.0081	>10.8*e*6 *ms*	0.5918	0.0072	>10.8*e*6 *ms*
Ours	0.6618	0.0019	106.88 *ms*	0.3697	7.807*e*−4	99.246 *ms*	0.4292	0.0022	91.146 *ms*

Table 5: Comparing the RMSE (root-mean-square error), PBMP (percentage of bad matching pixels), and mean run-time of the methods over the Middlebury dataset [31]. The standard deviation of the run-time is negligible.

values to both the left and right. In this case, the pattern of depth change can be sampled from either side depending on valid depth value availability within the same scene object. Assuming sufficient depth values exist to the left of the hole (by default, even if sufficient on the right also), we proceed as follows:

$$v_0(i) = v(i-1) + v(i-l) - v(i-l-1) \quad (3)$$

$$slope = \frac{v(i+l) - v(i-1)}{v_0(i) - v(i-l-1)} \quad (4)$$

$$v(i) = v(i-1) + [v(i-l) - v(i-l-1)] \times slope \quad (5)$$

To predict the missing depth values correctly considering the pattern of texture and relief, continuity between the valid values to the left and the right side of the hole is taken into account. The continuity coefficient (*slope*) is utilized to ensure that the predicted values plausibly bridge the depth values to the left and right of the hole. The pattern of change in the valid values is propagated rightward with each value being multiplied by *slope*, calculated by dividing the difference between the values surrounding the hole into the difference between the values surrounding the sample area (Figure 1 (Case 3) and Algorithm 1). The initial value of $v_0(i)$ and *slope* in Algorithm 1 are respectively calculated based on Eqns. 3 and 4. Within Algorithm 1, $v(i)$ is updated according to Eqn. 5. See the illustration of Eqns. 3, 4, and 5 terms in Figure 1 (Case 3).

***Case 4*:** as per Case 3, but such that the number of valid depth values to the left of the constrained hole is exactly the same as the length of the hole itself.

$$v_0(i) = v(i-1) + v(i-l+1) - v(i-l) \quad (6)$$

$$slope = \frac{v(i+l) - v_0(i)}{v_0(i) - v(i-l)} \quad (7)$$

$$v(i) = v(i-1) + [v(i-l+1) - v(i-l)] \times slope \quad (8)$$

The difference between this completion process and that of Case 3 is the same as the difference between Cases 1 and 2. The completion order and the *slope* coefficient are applied similarly to Case 3. The initial value of $v_0(i)$ and *slope* in Algorithm 1 are respectively calculated based on Eqns. 6 and 7. Within Algorithm 1, $v(i)$ is updated according to Eqn. 8. See the illustration of Eqns. 6, 7, and 8 terms in Figure 1 (Case 4).

***Case 5*:** where the constrained hole does not reach the leftmost or rightmost boundary of the object (as per Case 3) but the number of valid depth values to the left of the hole is smaller than the length of the hole itself, while sufficient valid depth values exist to the right of the hole for completion.

$$v_0(i) = v(i+1) + v(i+l) - v(i+l+1) \quad (9)$$

$$slope = \frac{v(i+1) - v(i-l)}{v(i+l+1) - v_0(i)} \quad (10)$$

$$v(i) = v(i+1) + [v(i+l) - v(i+l+1)] \times slope \quad (11)$$

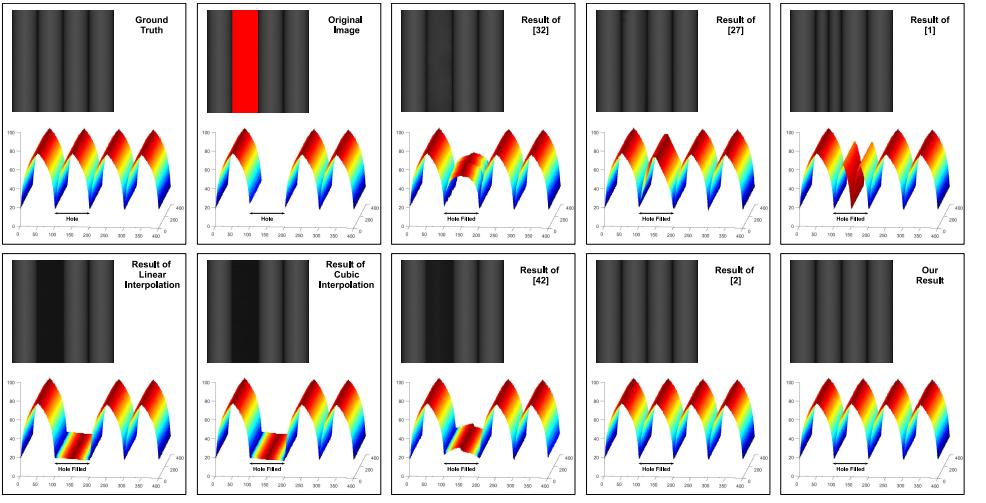

Figure 2: Comparison of proposed method against [1, 2, 29, 35, 45] and linear and cubic interpolation (synthetic test image with available ground truth depth).

Following a symmetric completion process to that of Case 3, the pattern of change in the valid depth values is propagated leftward as per Algorithm 1. The initial value of $v_0(i)$ and *slope* in Algorithm 1 are respectively calculated based on Eqns. 9 and 10. Within Algorithm 1, $v(i)$ is updated according to Eqn. 11. See the illustration of Eqns. 9, 10, and 11 terms in Figure 1 (Case 5).

***Case 6*:** as per Case 5, but such that the number of valid depth values to the right of the constrained hole is exactly the same as the length of the hole itself.

$$v_0(i) = v(i+1) + v(i+l-1) - v(i+l) \quad (12)$$

$$slope = \frac{v_0(i) - v(i-l)}{v(i+l) - v_0(i)} \quad (13)$$

$$v(i) = v(i+1) + [v(i+l-1) - v(i+l)] \times slope \quad (14)$$

Following a symmetric completion process to that of Case 4, the pattern of change in the valid depth values is propagated leftward as per Algorithm 1. The initial value of $v_0(i)$ and *slope* in Algorithm 1 are respectively calculated based on Eqns. 12 and 13. Within Algorithm 1, $v(i)$ is updated according to Eqn. 14. See the illustration of Eqns. 12, 13, and 14 terms in Figure 1 (Case 6).

***Case 7*:** where the constrained hole starts at the leftmost boundary edge of the scene object (symmetric to that of Case 1). Conversely, the number of valid values on the right of the hole is greater than the length of the hole itself.

$$v(i) = v(i+1) + [v(i+l) - v(i+l+1)] \times slope \quad (15)$$

Following a symmetric completion process to that of Case 1, the pattern of change in the valid depth values is propagated leftward as per Algorithm 1. Since no continuity is required, $slope = 1$. The initial value of $v(i)$ is zero, and this value is updated iteratively based on Eqn. 15. See the illustration of Eqn. 15 terms in Figure 1 (Case 7).

***Case 8*:** as per Case 7, but such that the number of valid depth values to the right of the constrained hole is exactly the same as the length of the hole itself.

$$v(i) = v(i+1) + [v(i+l-1) - v(i+l)] \times slope \quad (16)$$

The difference between this completion process and that of Case 7 is the same as the difference between Cases 1 and 2. The depth completion order and the *slope* coefficient are

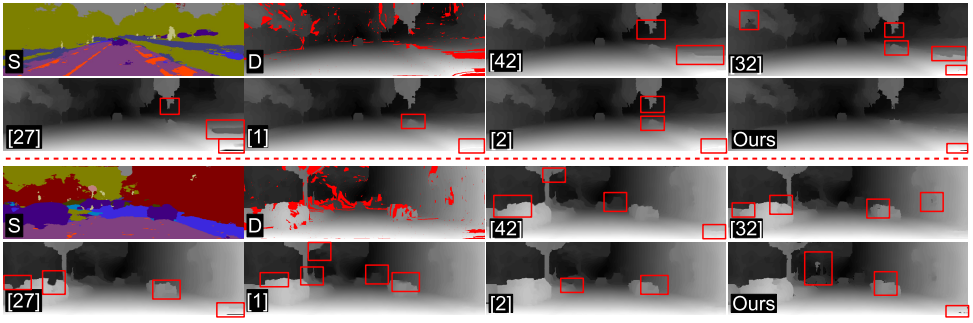

Figure 3: Exemplar results on the KITTI dataset [27]. *S* denotes the segmented images [3] and *D* the original (unfilled) disparity maps. Results are compared with [1, 2, 29, 35, 45]. Results of cubic and linear interpolations are omitted due to space.

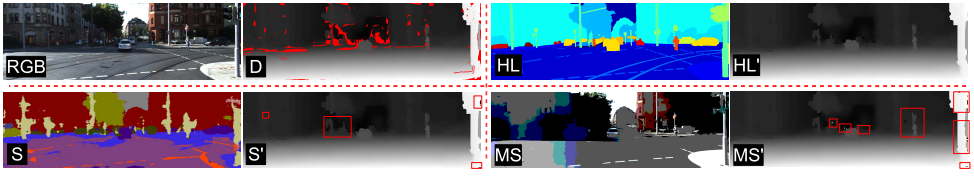

Figure 4: Comparison of the proposed method using different initial segmentation techniques on the KITTI dataset [27]. Original color and disparity image (top-left), results with manual labels (top-right), results with SegNet [3] (bottom-left) and results with mean-shift [26] (bottom-right).

applied similarly to Case 7. Since no continuity is required, $slope = 1$. In Algorithm 1, the initial value of $v(i)$ is zero , and this value is updated iteratively based on Eqn. 16. See the illustration of Eqn. 16 terms in Figure 1 (Case 8).

***Case 9*:** where the constrained hole extends to the rightmost pixel within the object (similar to Cases 1 and 2), but we cannot employ a non-parametric approach (as per Cases 1 and 2) because the number of valid depth values to the left of the hole is smaller than the length of the hole itself. As a result, there is not enough information to accurately fill these holes. Instances of these cases are left unfilled if identified during the scan in progress. In subsequent scans, many of these unresolvable (Case 9) patterns are broken due to the use of alternating row-wise and column-wise scan passes (resulting in an alternative resolvable case instance). For Case 9 instances that are not resolved after all three image passes, simple (cubic) interpolation is used (in an insignificant number of cases, Table 1).

***Case 10*:** where the constrained hole extends to the leftmost pixel within the scene object (similar to Cases 7 and 8), but again we cannot employ a non-parametric approach (as per Cases 7 and 8) because the number of valid depth values to the right of the hole is smaller than the length of the hole itself. As a result, there is again not enough information to accurately fill these holes, and we proceed as per Case 9.

***Case 11*:** where the constrained hole is located in the middle of an object but with insufficient valid depth values to the left and right side to facilitate non-parametric filling. Again, there is not enough information to accurately fill these holes, and we proceed as per Case 9.

***Case 12*:** where the constrained hole spans over the entire length of the scene object, (i.e. no depth is available for an object known to be present in the scene from the semantically segmented color image) making it the most challenging case of all. For instances of this case not resolved within the three scan passes (row-wise, column-wise, secondary row-wise),

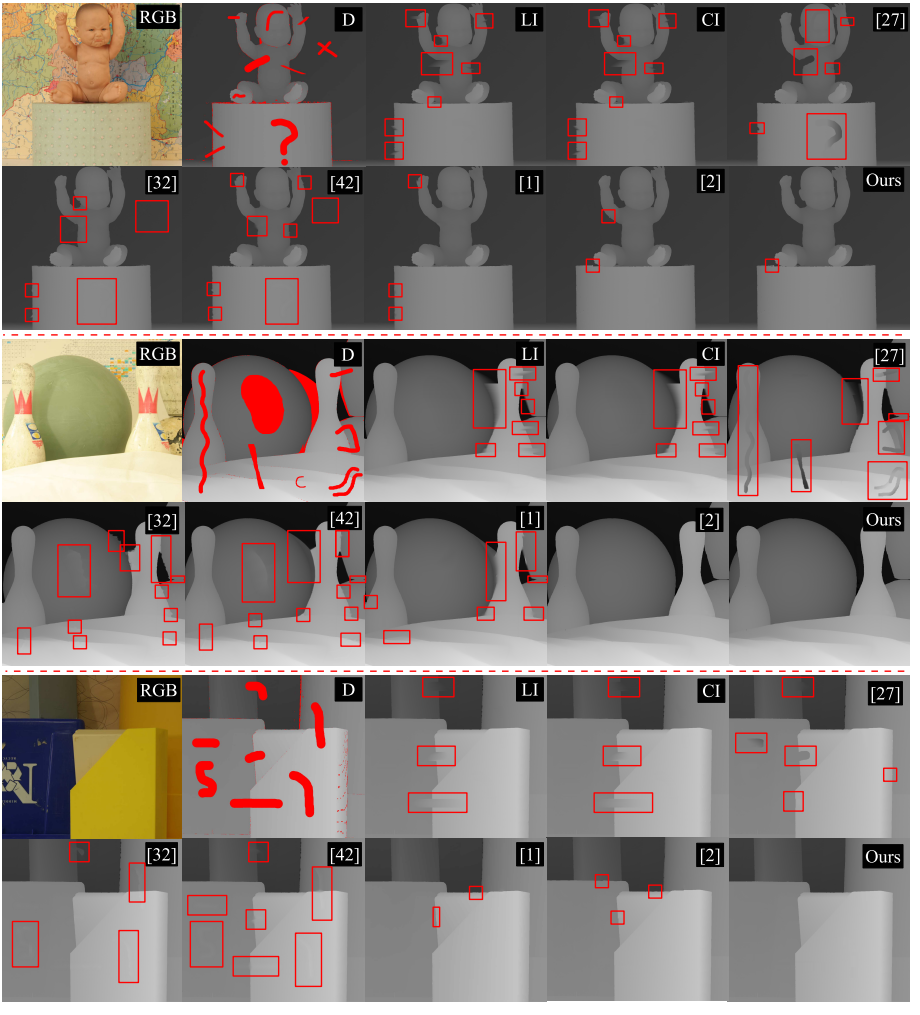

Figure 5: Exemplar results on the Middlebury dataset [31]. *RGB* denotes the original color images and *D* the original (unfilled) depth maps. Results are compared with [1, 2, 29, 35, 45] and linear and cubic interpolation methods.

a clear ambiguity exists as there is no valid depth information available for the object at all. As Table 1 illustrates, this is an incredibly rare occurrence in practice, and the hole is best left uncompleted rather than using invalid or implausible values (as per other work, [1, 2, 13, 14, 18, 22, 29, 34, 35, 37, 39, 40, 41, 45, 47, 50, 51]). Table 1 illustrates the typical occurrence frequency of the cases (1-12) on the KITTI dataset [27] (using [49] for depth estimation). As seen in Table 1, less than 2% of hole occurrences cannot be completed using our three-pass approach (row-wise, column-wise, secondary row-wise) through the resolution outlined. This includes the challenging Case 12, which cannot be accurately filled due to the lack of surrounding valid depth values. Although our three pass processing of these 12 cases noticeably uses no explicit inter-row/column support regions (from adjacent row/columns) as may be ordinarily expected, this is in fact implicit in our formulation based on the use of the prior region-based scene segmentation (as illustrated in Section 4), which inherently provides semantically defined support regions.

4 Experimental Results

With the asymptotic runtime of $O(n)$ for n image pixels, the proposed method is comparable to simple interpolation methods in complexity but with accuracy exceeding that of more complex methods [1, 2, 29, 35, 45]. The approach is first tested using a synthetically generated depth image (Figure 2). This image contains steep curves and sharp peaks to simulate exaggerated texture to evaluate the performance in presence of surface relief within the image (under a single scene object assumption, with no need for prior segmentation). Here, Gaussian noise ($mean = 0$, $variance = 0.0001$) is added to the depth image to avoid completely smooth surfaces, and a topological color scale is used to guide methods that require additional color image input ([29, 35]), and to aid visualization of the final result.

The superiority of the proposed method is clearly seen in Figure 2. Additionally, the root-mean-square error (RMSE) and the percentage of bad pixels produced by the proposed method are far smaller than comparators, as seen in Table 3.

Figure 3 demonstrates the results of the proposed method in comparison with others when applied to examples from the KITTI dataset [27] (resolution, 1242×375). Depth is calculated using [49] with significant disparity speckles filtered out and SegNet [3] is used to perform the initial semantic scene understanding. The proposed method results in sharper images with no additional artefacts (Figure 3) and performs more efficiently than comparator approaches (see Table 4).

As previously discussed, the initial segmentation step can indeed be performed using any technique with the efficacy of results depending on the accuracy of this segmentation. Figure 4 compares the results of our approach obtained through the use varying segmentation methods. When a manually labeled image is used (ground truth, [48]), the results are more accurate than when SegNet [3] or mean-shift [20, 21, 26] segmentation is employed.

We also utilize the Middlebury dataset [31] to provide additional qualitative and quantitative evaluation. Figure 5 demonstrates that the proposed method generates more plausible results without invalid outliers, blurring, jagging or other artefacts than comparator approaches. Table 5 provides quantitative evaluation of the proposed approach against the same comparator set. As shown in Table 5, the method is faster (real time, excluding segmentation) and has a smaller root-mean-square error and fewer bad pixels [42] than comparators. Experiments were performed on a 2.30GHz CPU using 8GB of memory (Tables 3, 5 and 4).

5 Conclusion

In this paper, the problem of depth image completion is addressed with efficiency, and attention to surface (relief) detail accuracy, with reference to a prior object-wise scene labeling. This first step requires an accurate semantic segmentation over an accompanying color image, which is commonly available from contemporary sensing arrangements, to facilitate depth completion on an object-wise basis. Missing depth values are subsequently filled via a three-pass non-parametrically driven approach, using a grammar of twelve discrete completion case occurrences. Our evaluation demonstrates that while the efficiency of the proposed method is comparable to simple interpolation methods, the plausibility and statistical relevance of the depth filled results compete against the accuracy of contemporary depth-filling approaches in the field. Fine depth surface detail and relief texture is preserved within a highly efficient framework driven by recent and ongoing advances in scene labeling.

References

[1] Pablo Arias, Gabriele Facciolo, Vicent Caselles, and Guillermo Sapiro. A variational framework for exemplar-based image inpainting. *Int. J. Computer Vision*, 93(3):319–347, 2011.

[2] Amir Atapour-Abarghouei, Gregoire Payen de La Garanderie, and Toby P. Breckon. Back to butterworth - a fourier basis for 3d surface relief hole filling within rgb-d imagery. In *Int. Conf. Pattern Recognition*. IEEE, 2016.

[3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[4] Seung-Hwan Baek, Inchang Choi, and Min H Kim. Multiview image completion with space structure propagation. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 488–496, 2016.

[5] Yuriy Berdnikov and Dmitriy Vatolin. Real-time depth map occlusion filling and scene background restoration for projected-pattern based depth cameras. In *Graphic Conf. IETP*, 2011.

[6] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Conf. Computer graphics and interactive techniques*, pages 417–424, 2000.

[7] Serge Beucher and Fernand Meyer. The morphological approach to segmentation: the watershed transformation. *Optical Engineering - New York*, 34:433–433, 1992.

[8] Toby P. Breckon and Robert B. Fisher. Amodal volume completion: 3d visual completion. *Computer Vision and Image Understanding*, 99(3):499–526, 2005.

[9] Toby P. Breckon and Robert B. Fisher. 3D surface relief completion via non-parametric techniques. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 30(12):2249 – 2255, 2008.

[10] Toby P. Breckon and Robert B. Fisher. A hierarchical extension to 3D non-parametric surface relief completion. *Pattern Recognition*, 45:172–185, September 2012.

[11] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *IEEE Conf. Computer Vision and Pattern Recognition*, volume 2, pages 60–65, 2005.

[12] Aurélie Bugeau, Marcelo Bertalmío, Vicent Caselles, and Guillermo Sapiro. A comprehensive framework for image inpainting. *IEEE Trans. Image Processing*, 19(10): 2634–2645, 2010.

[13] Massimo Camplani and Luis Salgado. Adaptive spatio-temporal filter for low-cost camera depth maps. In *Int. Conf. Emerging Signal Processing Applications*, pages 33–36. IEEE, 2012.

[14] Massimo Camplani and Luis Salgado. Efficient spatio-temporal hole filling strategy for kinect depth maps. In *IS&T/SPIE Electronic Imaging*, pages 82900E–82900E, 2012.

[15] Pedro Cavestany, Antonio L. Rodriguez, Humberto Martinez-Barbera, and Toby P. Breckon. Improved 3d sparse maps for high-performance structure from motion with low-cost omnidirectional robots. In *Int. Conf. Image Processing*, pages 4927–4931. IEEE, 2015.

[16] T Chan and J Shen. Mathematical models for local deterministic in-paintings. Technical report, Technical Report CAM TR 00-11, 2000.

[17] Chongyu Chen, Jianfei Cai, Jianmin Zheng, Tat Jen Cham, and Guangming Shi. Kinect depth recovery using a color-guided, region-adaptive, and depth-selective framework. *ACM Trans. Intelligent Systems and Technology*, 6(2):12, 2015.

[18] Li Chen, Hui Lin, and Shutao Li. Depth image enhancement for kinect using region growing and bilateral filter. In *Int. Conf. Pattern Recognition*, pages 3070–3073. IEEE, 2012.

[19] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[20] Yizong Cheng. Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(8):790–799, 1995.

[21] Dorin Comaniciu and Peter Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. pattern analysis and machine intelligence*, 24(5):603–619, 2002.

[22] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Processing*, 13(9): 1200–1212, 2004.

[23] Ismaël Daribo and Hideo Saito. A novel inpainting-based layered depth video for 3dtv. *IEEE Trans. Broadcasting*, 57(2):533–541, 2011.

[24] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *Int. Conf. Computer Vision*, pages 1033–1038. IEEE, 1999.

[25] Deliang Fu, Yin Zhao, and Lu Yu. Temporal consistency enhancement on depth sequences. In *Picture Coding Symposium*, pages 342–345. IEEE, 2010.

[26] Keinosuke Fukunaga and Larry Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. *IEEE Trans. information theory*, 21 (1):32–40, 1975.

[27] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *Int. J. Robotics Research*, 2013.

[28] Oliver K. Hamilton and Toby P. Breckon. Generalized dynamic object removal for dense stereo vision based scene mapping using synthesised optical flow. In *Int. Conf. Image Processing*, pages 3439–3443. IEEE, 2016.

[29] Daniel Herrera, Juho Kannala, Janne Heikkilä, et al. Depth map inpainting under a second-order smoothness prior. In *Scandinavian Conference on Image Analysis*, pages 555–566. Springer, 2013.

[30] Alexandre Hervieu, Nicolas Papadakis, Aurélie Bugeau, Pau Gargallo, and Vicent Caselles. Stereoscopic image inpainting: distinct depth maps and images inpainting. In *Int. Conf. Pattern Recognition*, pages 4101–4104. IEEE, 2010.

[31] Heiko Hirschmuller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2007.

[32] Christopher J. Holder, Toby P. Breckon, and Xiong Wei. From on-road to off: Transfer learning within a deep convolutional neural network for segmentation and classification of off-road scenes. In *European Conf. Computer Vision*, pages 149–162. Springer, 2016.

[33] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[34] Sung-Yeol Kim, Ji-Ho Cho, Andreas Koschan, and Mongi A Abidi. Spatial and temporal enhancement of depth images captured by a time-of-flight depth sensor. In *Int. Conf. Pattern Recognition*, pages 2358–2361. IEEE, 2010.

[35] Junyi Liu, Xiaojin Gong, and Jilin Liu. Guided inpainting and filtering for kinect depth maps. In *Int. Conf. Pattern Recognition*, pages 2055–2058. IEEE, 2012.

[36] Shaoguo Liu, Ying Wang, Jue Wang, Haibo Wang, Jixia Zhang, and Chunhong Pan. Kinect depth restoration via energy minimization with tv 21 regularization. In *Int. Conf. Image Processing*, pages 724–724. IEEE, 2013.

[37] Si Lu, Xiaofeng Ren, and Feng Liu. Depth enhancement via low-rank matrix completion. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 3390–3397, 2014.

[38] Sergey Matyunin, Dmitriy Vatolin, Yury Berdnikov, and Maxim Smirnov. Temporal filtering for depth maps generated by kinect depth camera. In *3DTV Conference*, pages 1–4. IEEE, 2011.

[39] Quang H Nguyen, Minh N Do, and Sanjay J Patel. Depth image-based rendering from multiple cameras with 3d propagation algorithm. In *Int. Conf. Immersive Telecommunications*, page 6, 2009.

[40] Fei Qi, Junyu Han, Pengjin Wang, Guangming Shi, and Fu Li. Structure guided fusion for depth map inpainting. *Pattern Recognition Letters*, 34(1):70–76, 2013.

[41] Christian Richardt, Carsten Stoll, Neil A Dodgson, Hans-Peter Seidel, and Christian Theobalt. Coherent spatiotemporal filtering, upsampling and rendering of rgbz videos. In *Computer Graphics Forum*, volume 31, pages 247–256, 2012.

[42] Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Computer Vision*, 47(1-3):7–42, 2002.

[43] Sunando Sengupta, Eric Greveson, Ali Shahrokni, and Philip HS Torr. Urban 3d semantic modelling using stereo vision. In *Int. Conf. Robotics and Automation*, pages 580–585. IEEE, 2013.

[44] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[45] Alexandru Telea. An image inpainting technique based on the fast marching method. *Graphics Tools*, 9(1):23–34, 2004.

[46] Liang Wang, Hailin Jin, Ruigang Yang, and Minglun Gong. Stereoscopic inpainting: Joint color and depth completion from stereo images. In *IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[47] Zhongyuan Wang, Jinhui Hu, ShiZheng Wang, and Tao Lu. Trilateral constrained sparse representation for kinect depth hole filling. *Pattern Recognition Letters*, 65: 95–102, 2015.

[48] Philippe Xu, Franck Davoine, Jean-Baptiste Bordes, Huijing Zhao, and Thierry Denœux. Multimodal information fusion for urban scene understanding. *Machine Vision and Applications*, 27(3):331–349, 2016.

[49] Koichiro Yamaguchi, David McAllester, and Raquel Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *European Conference on Computer Vision*, pages 756–771. Springer, 2014.

[50] Jingyu Yang, Xinchen Ye, Kun Li, Chunping Hou, and Yao Wang. Color-guided depth recovery from rgb-d data using an adaptive autoregressive model. *IEEE Trans. Image Processing*, 23(8):3443–3458, 2014.

[51] Liang Zhang, Peiyi Shen, Shu'e Zhang, Juan Song, and Guangming Zhu. Depth enhancement with improved exemplar-based inpainting and joint trilateral guided filtering. In *Int. Conf. Pattern Recognition*, pages 4102–4106. IEEE, 2016.